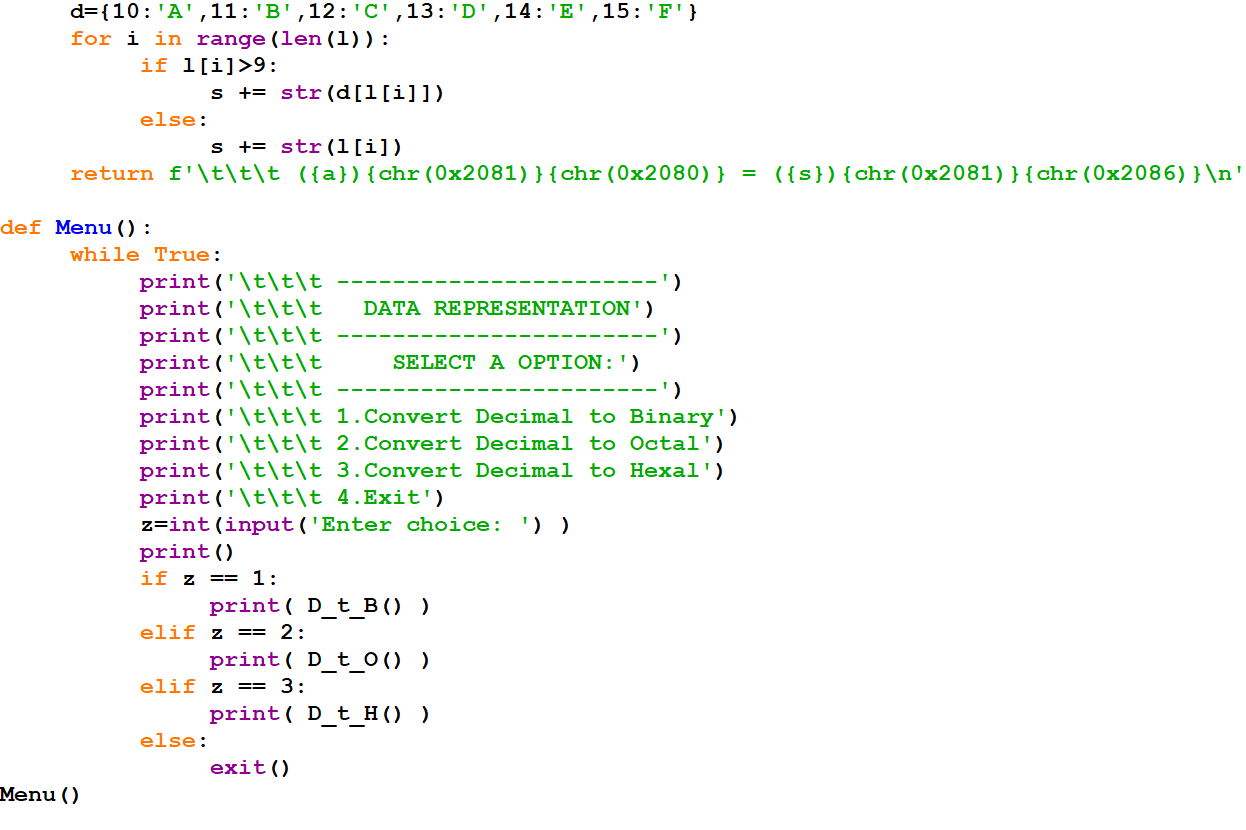
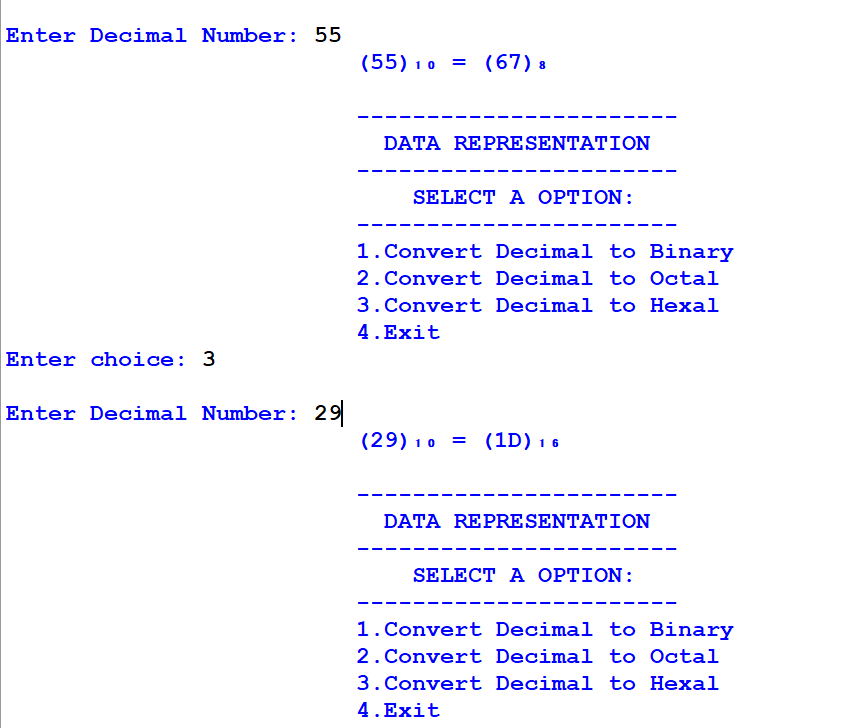
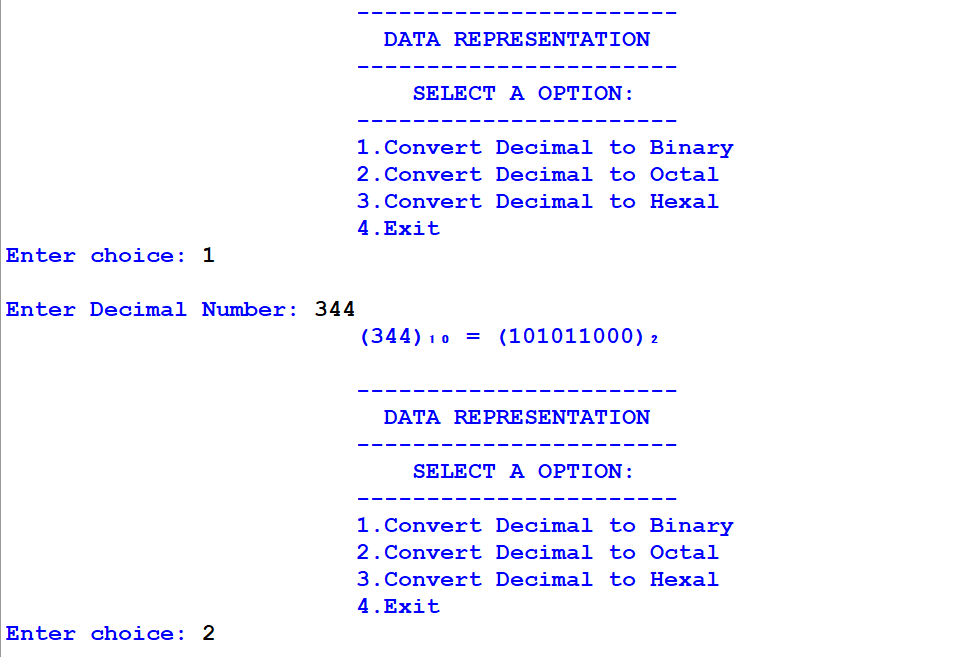
1. NUMBER SYSTEM

**Q.) Create a menu driven program to convert a decimal number to it’s binary, octal, and hexagonal equivalents.**

**INPUT**



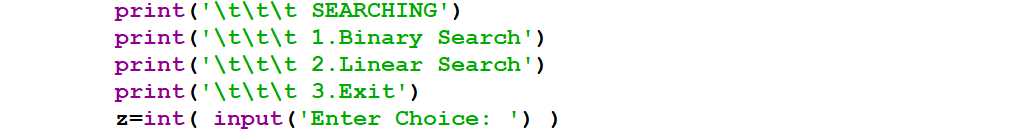
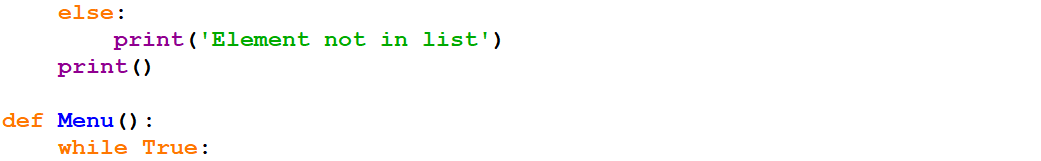
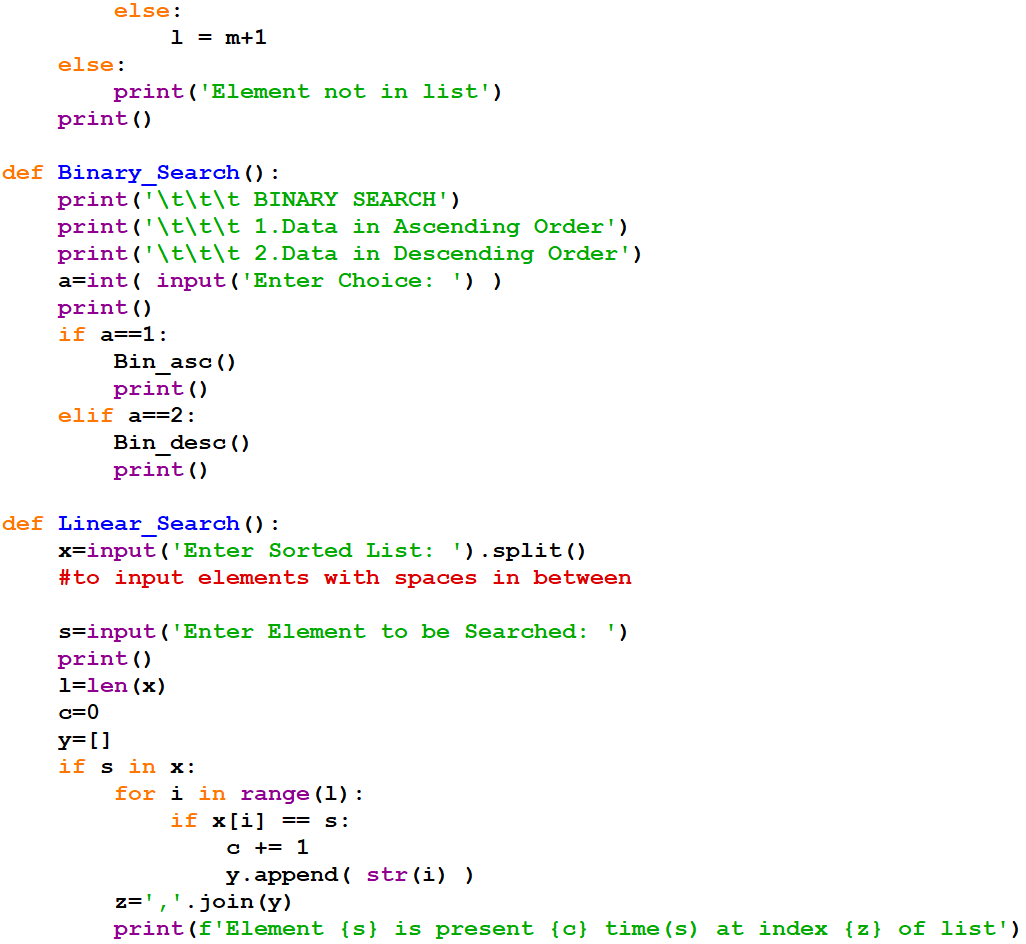
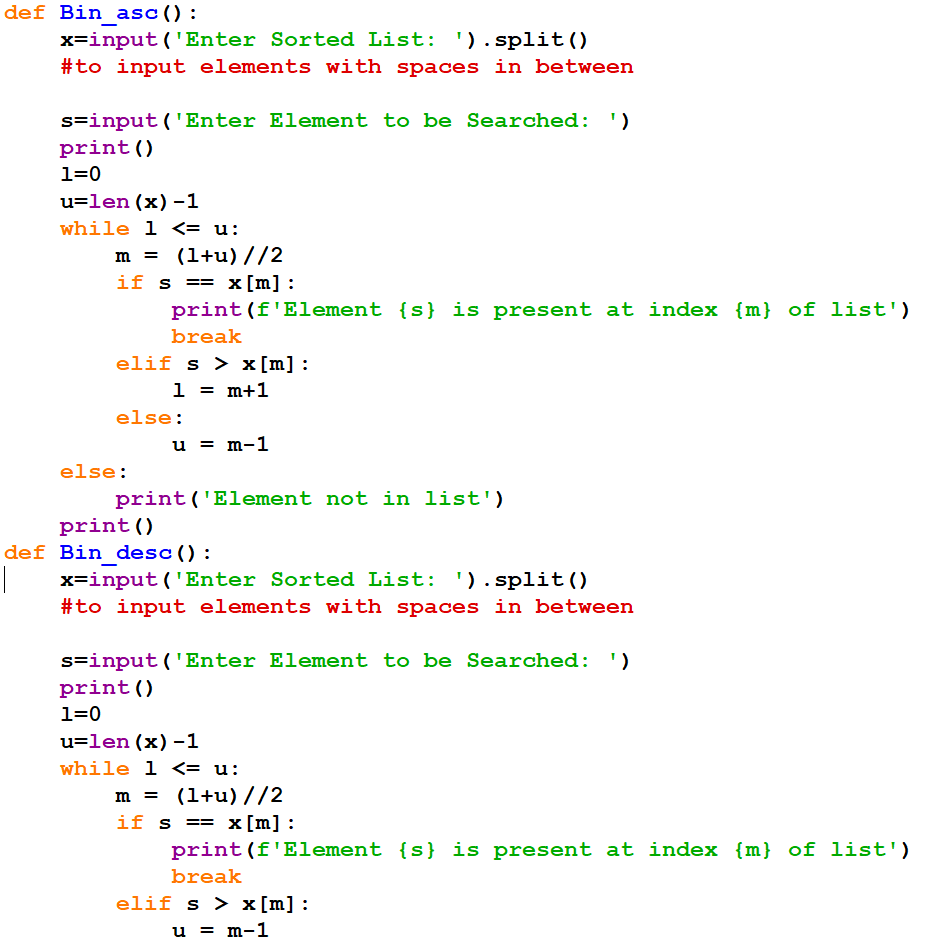
**OUTPUT**



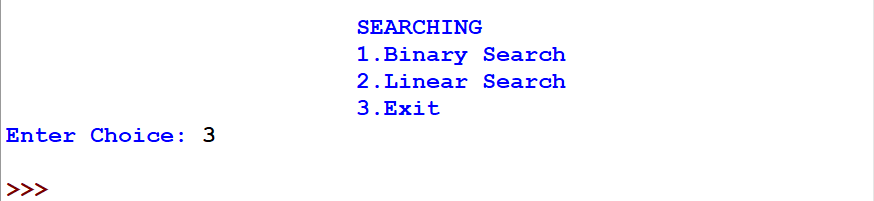
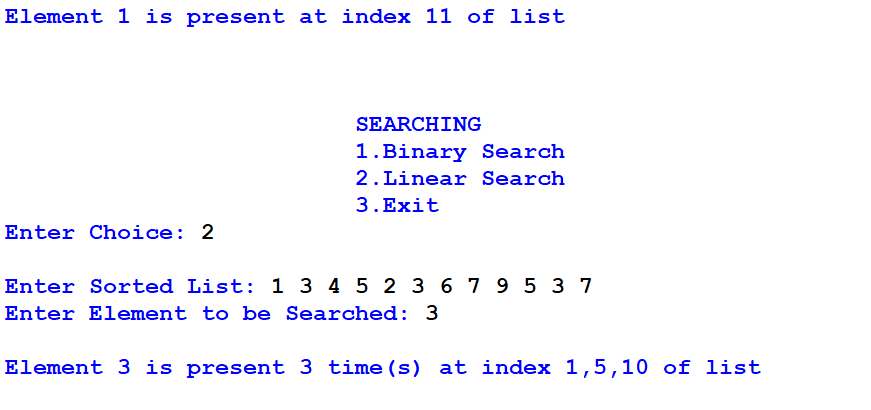
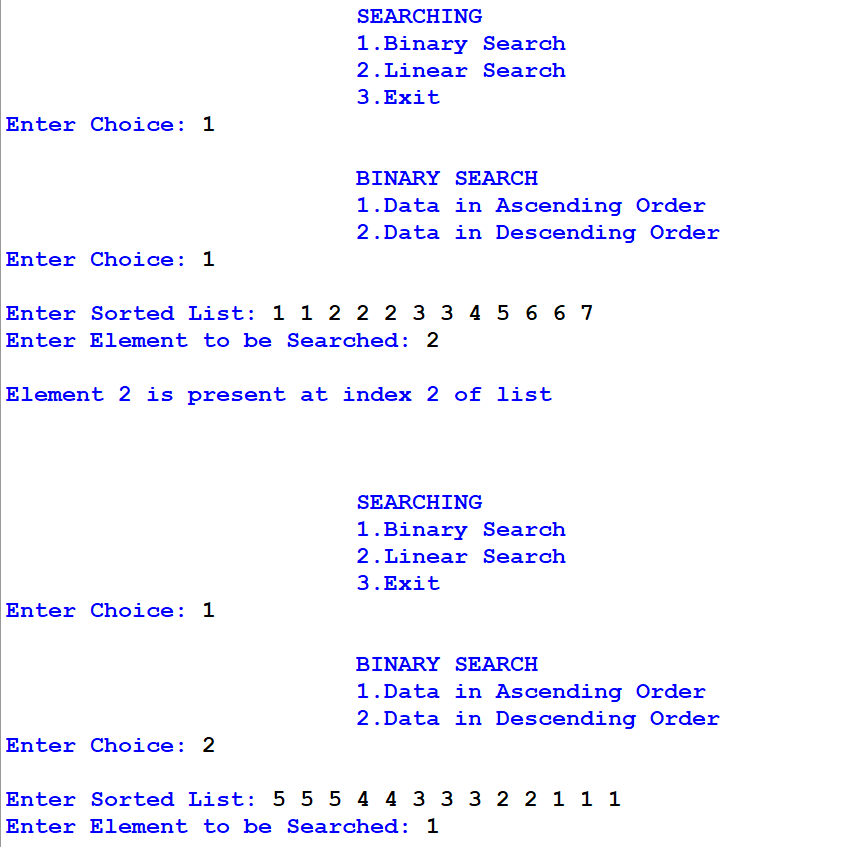
2. SEARCHING

**Q.) Create a menu driven program to search an element in a sorted list.**

**INPUT**



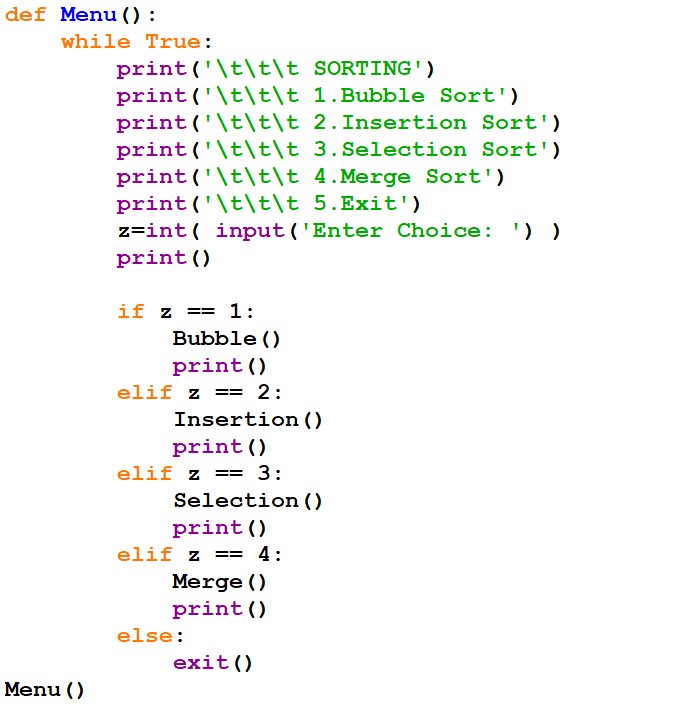
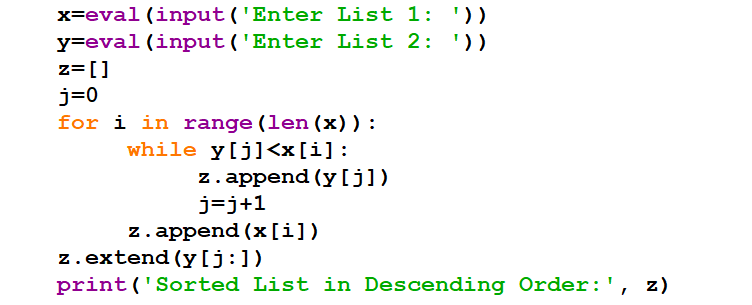
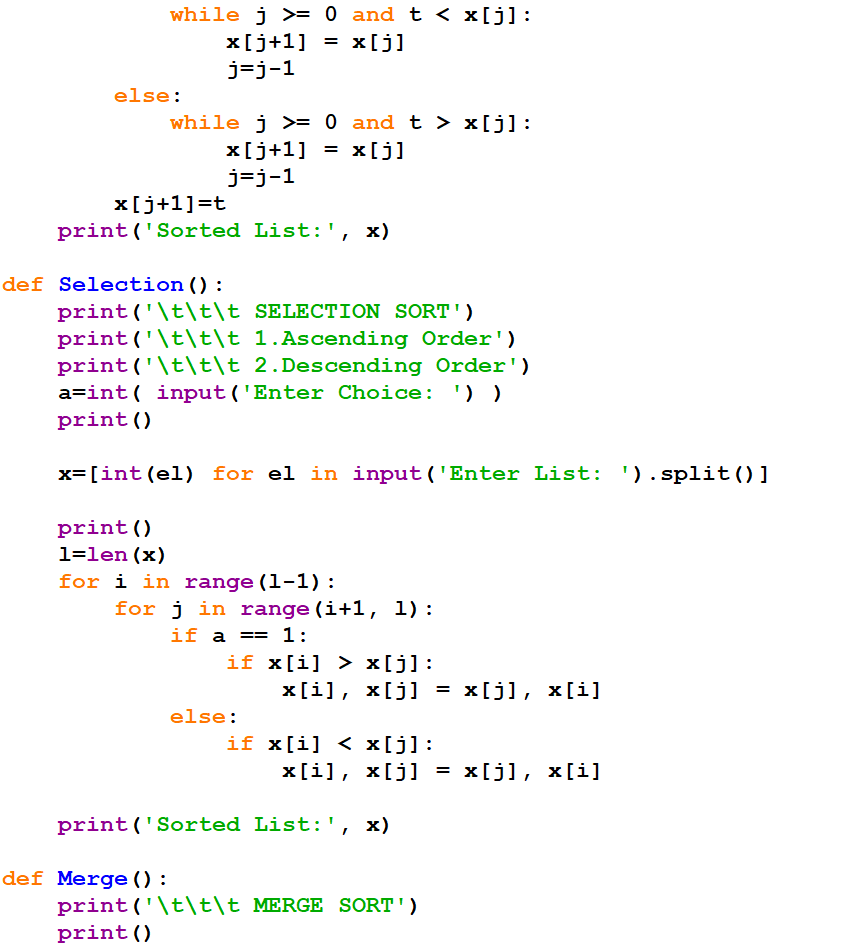
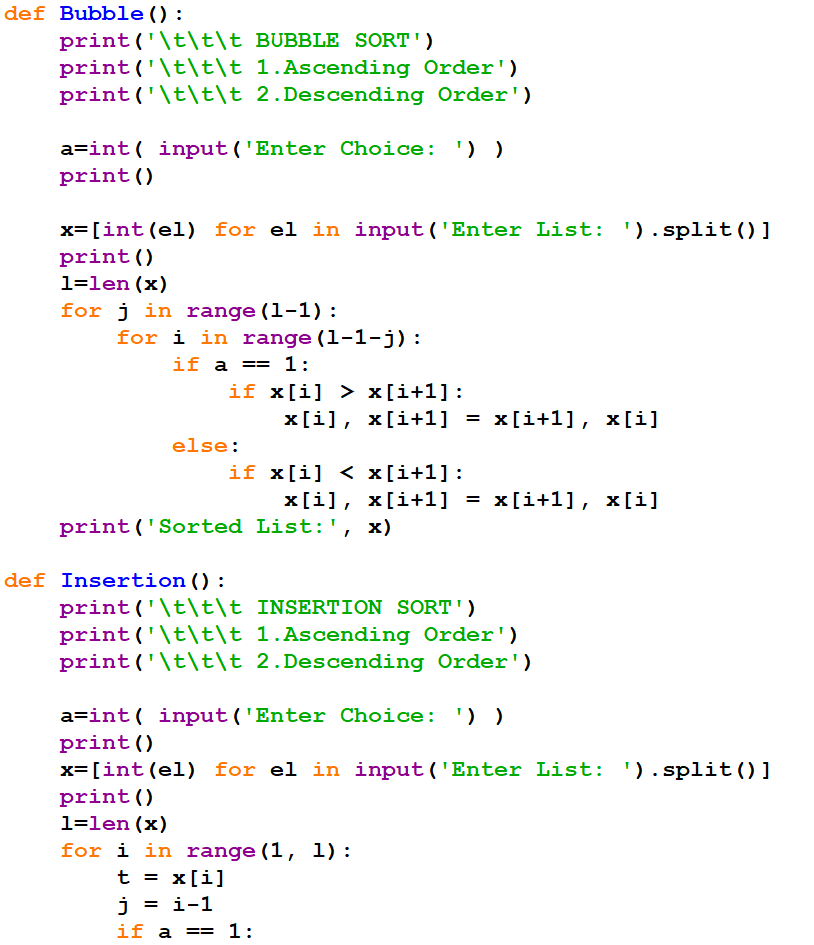
**OUTPUT**



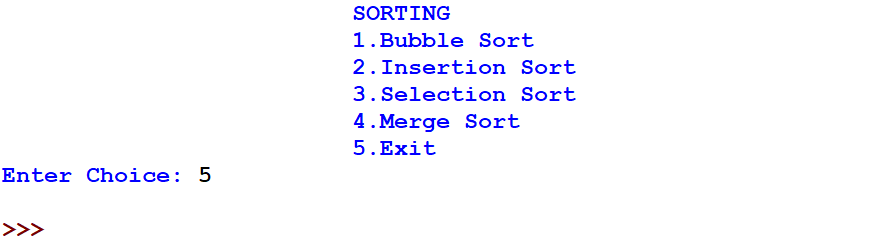
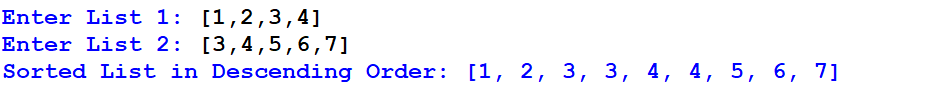
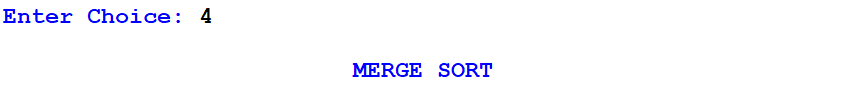
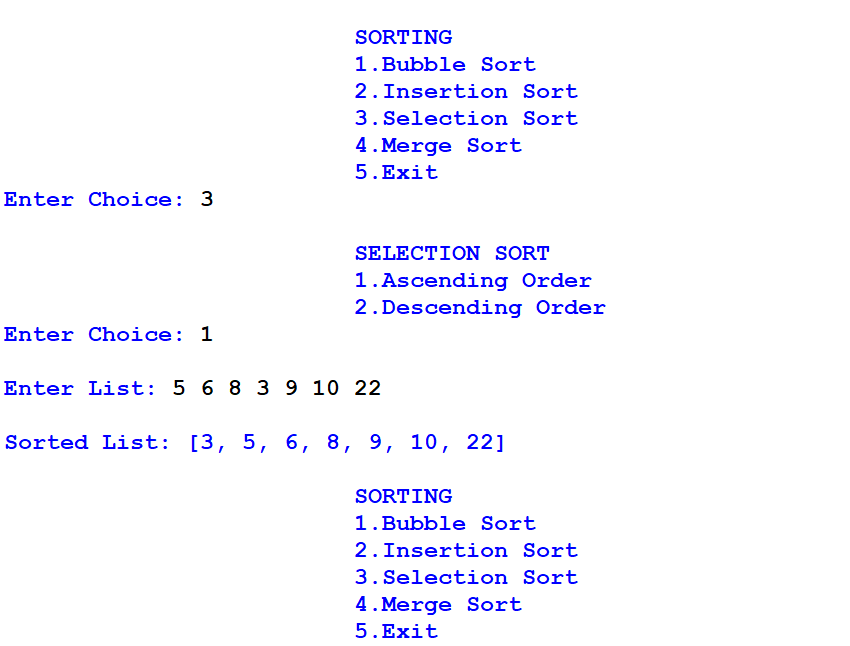
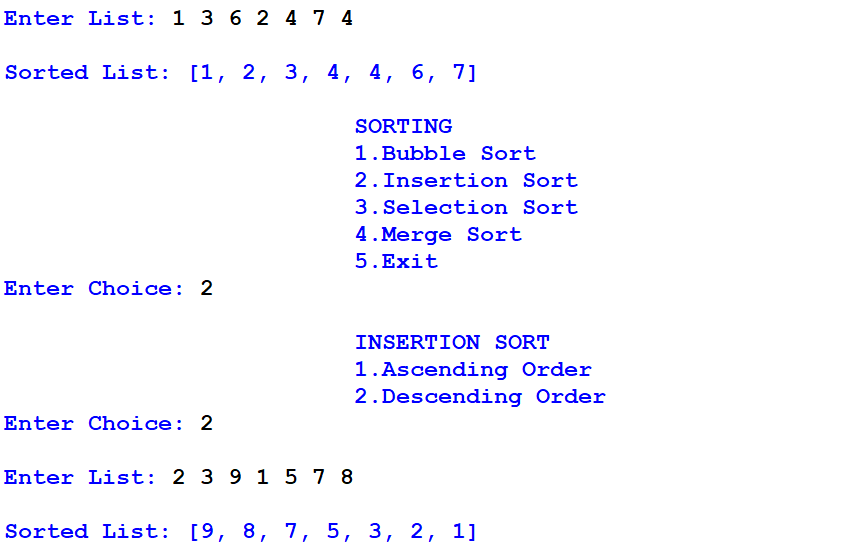
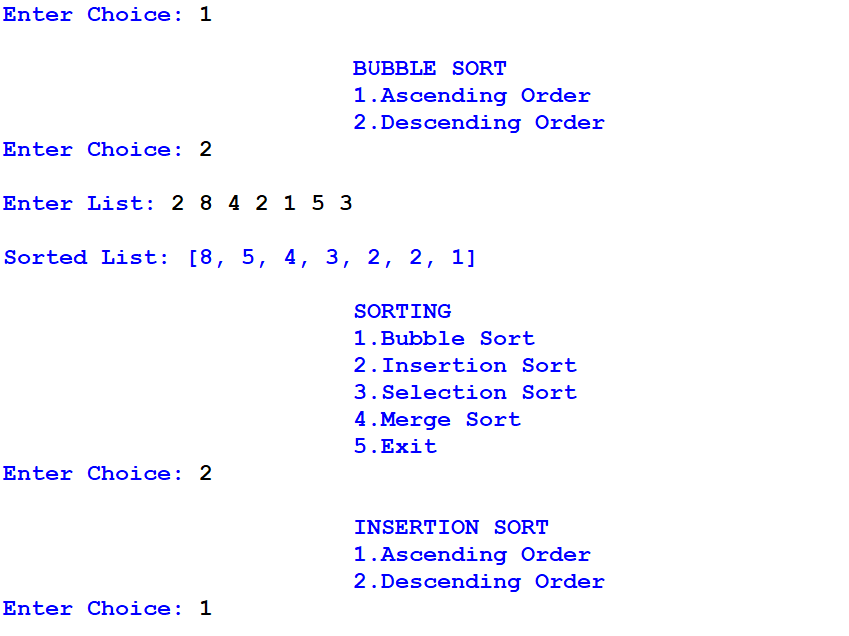
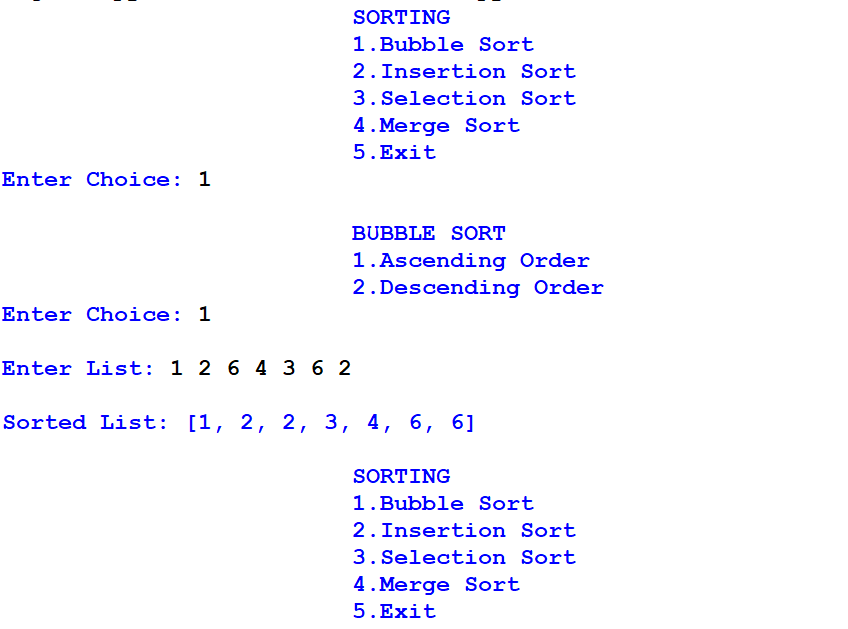
3. SORTING

**Q.) Create a menu driven program to sort the elements of a given list using different algorithms.**

**INPUT**



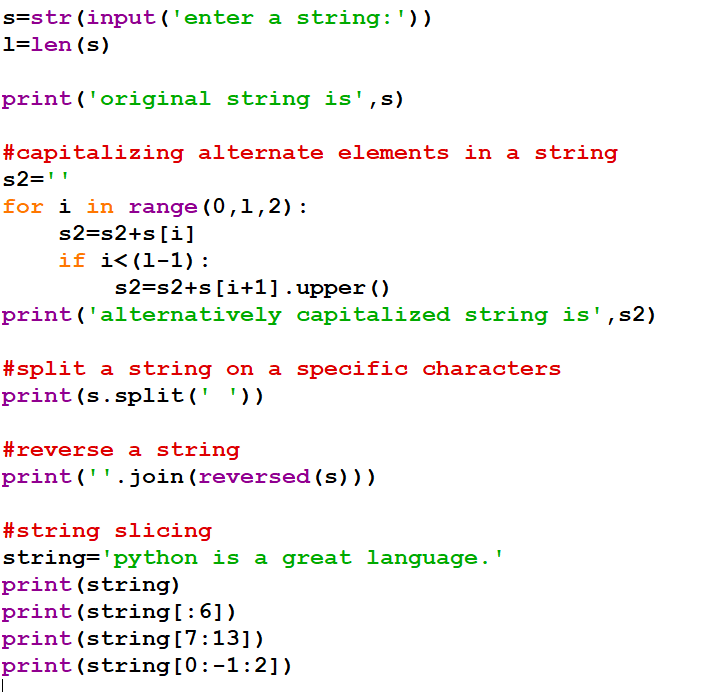
**OUTPUT**



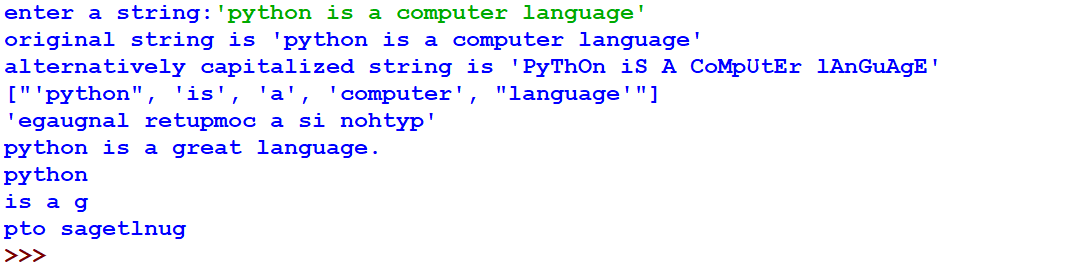
4. STRING HANDLING

**Q.) Write a program that inputs and reads a string and performs various inbuilt python string manipulation methods.**

**INPUT**



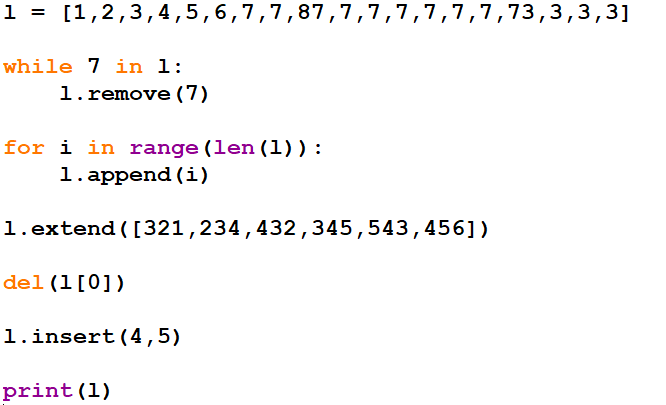
**OUTPUT**



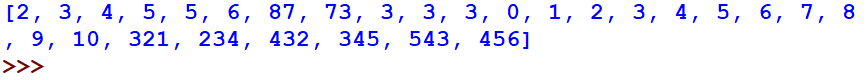
5. LIST HANDLING

**Q.) Write a program that inputs and reads a list and performs various inbuilt python list manipulation methods.**

**INPUT**



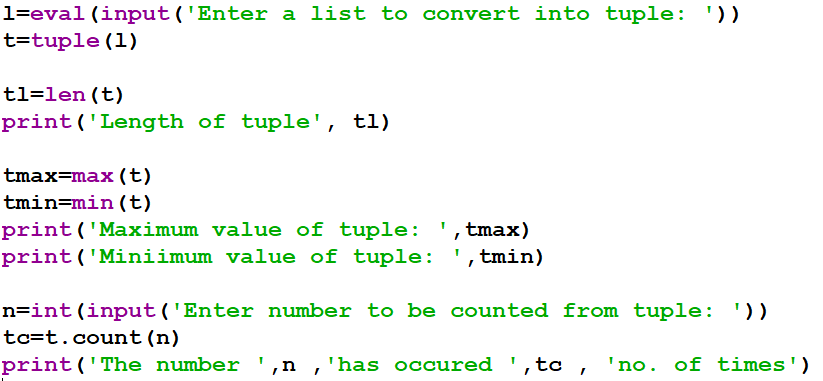
**OUTPUT**



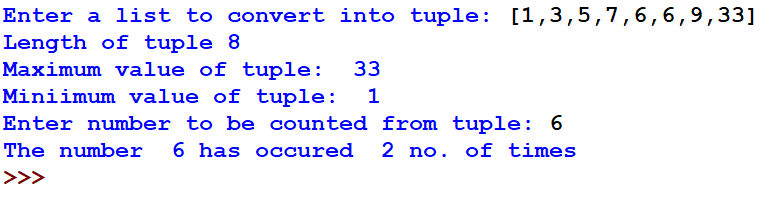
6. TUPLE HANDLING

**Q.) Write a program that inputs and reads a tuple and performs various inbuilt python tuple manipulation methods.**

**INPUT**



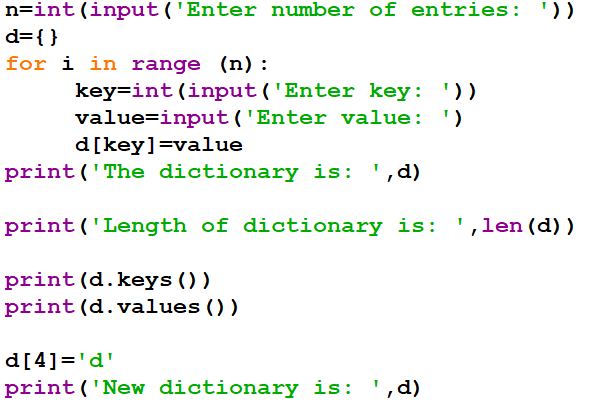
**OUTPUT**



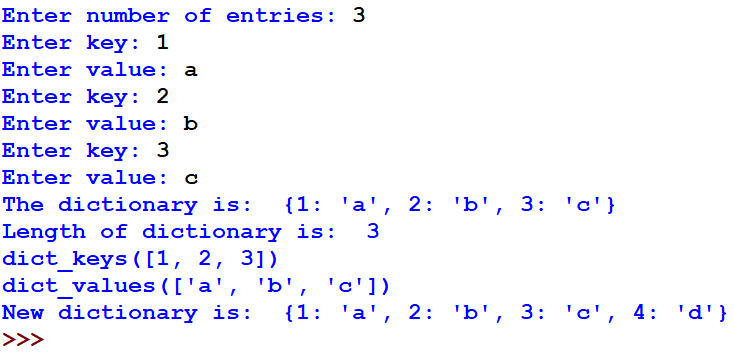
7. DICTIONARY HANDLING

**Q.) Write a program that inputs and reads a dictionary and performs various inbuilt python dictionary manipulation methods.**

**INPUT**



**OUTPUT**



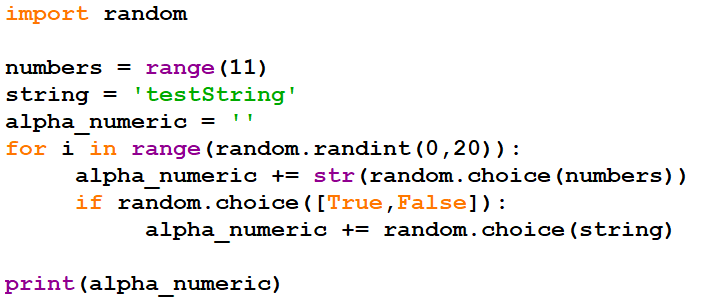
8. LOGIC BASED PROGRAM ON STRING/LIST/TUPLE/DICTIONARY

9. FLOOR DIVISION / MODULUS

10. OPERATORS IN PYTHON

11. RANDOM

**INPUT**



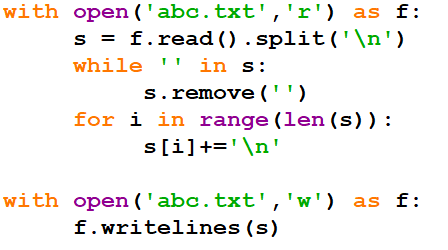
**OUTPUT**



12. FILE HANDLING (TEXT MODE)

**--> removing all empty lines**

**INPUT**



**OUTPUT**

Initially:

file

handling

text

Finally:

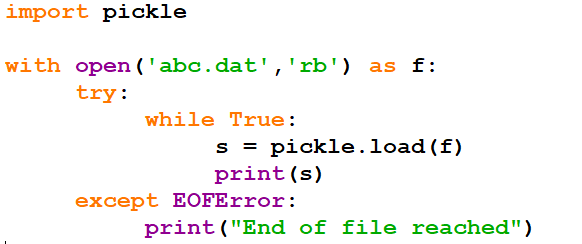
file

handling

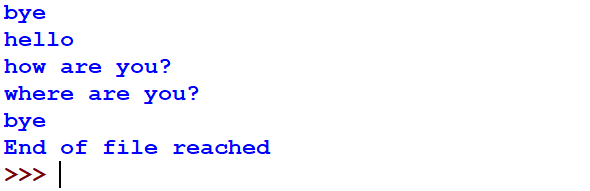
text

13. FILE HANDLING (BINARY MODE)

**INPUT**

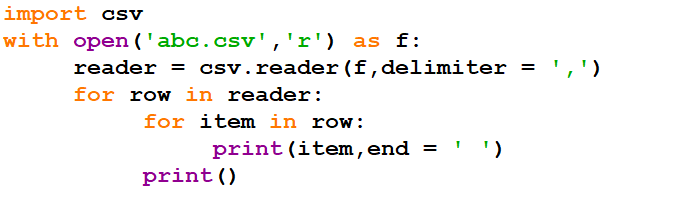


**OUTPUT**



14. FILE HANDLING (CSV)

**INPUT**



**OUTPUT**



15. USER DEFINED FUNCTION

16. MYSQL QUERIES

**Program 1*:*** *Write a SQL program to create a table with suitable fields*

*Code:*

**CREATE TABLE jobs (**

**JOB\_ID integer NOT NULL UNIQUE PRIMARY KEY,**

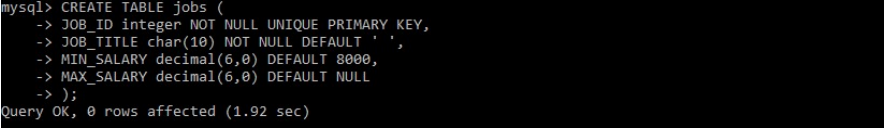
**JOB\_TITLE varchar(35) NOT NULL DEFAULT ' ',**

**MIN\_SALARY decimal(6,0) DEFAULT 8000,**

**MAX\_SALARY decimal(6,0) DEFAULT NULL**

**);**

*Output:*



**Program 2:** *Write a MySQL program to enter three records at once.*

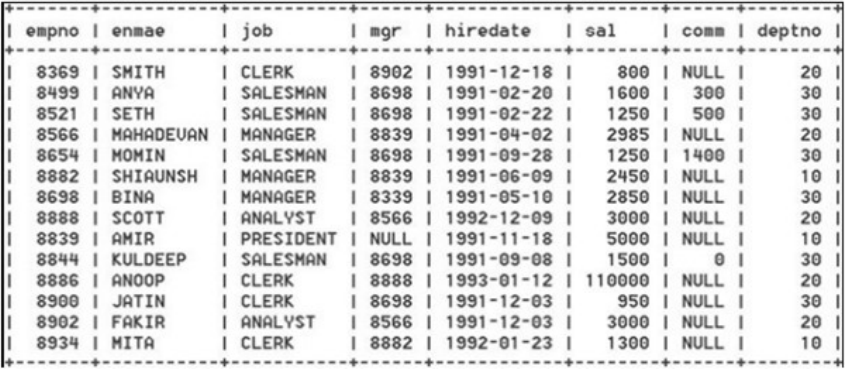
*Code:*

**insert into jobs values(281,'Engineer',24435,291198),(991,'Engineer',246645 ,291198),(721,'Engineer',287645,291348);**

*Output:*



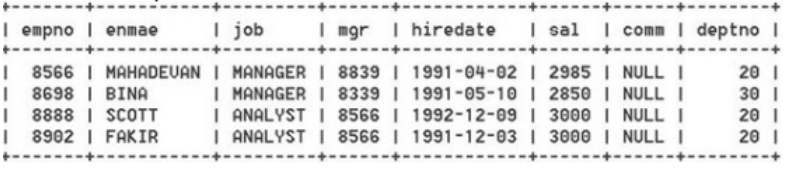
*Consider the table empl given below and run the following Queries.*



*1. List the details of those employees whose annual salary is between*

*25000 and 40000.*

**SELECT \* FROM empl where sal BETWEEN 2500 AND 4000;**



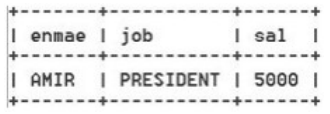
*2. Display the name of employees whose name contains ‘A’ as the 4th alphabet.*

**SELECT ename FROM empl WHERE ename like ‘\_\_\_a%’;**



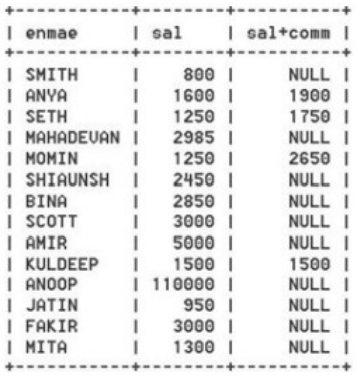
*3. Display Name, Job and Salary of employees who do not have a manager.*

**SELECT ename, job, sal FROM empl WHERE mgr IS NULL;**



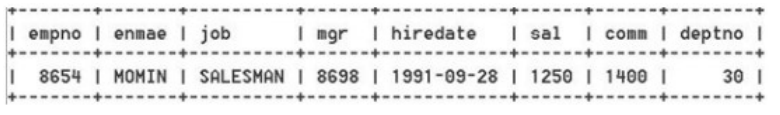
*4. Display Name, Salary and salary added with commision.*

**SELECT ename, sal, sal+comm FROM empl;**

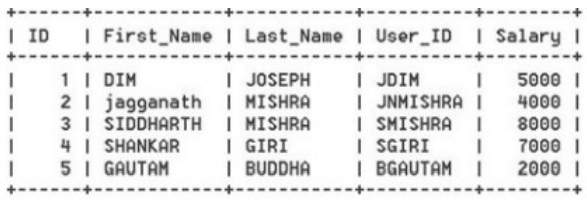


*5. Display details of employees who earn more commission than their salaries.*

**SELECT \* FROM empl WHERE comm>sal;**



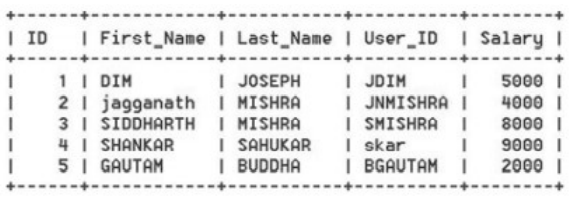
*Consider the table EMPLOYEE given below and run the following Queries.*



*1. For record with ID=4 update record with last Name, User ID and Salary.*

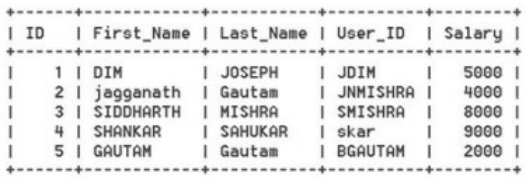
**UPDATE employe SET Last\_Name=’SAHUKAR’,User\_ID=’skar’,Salary=9000**

**WHERE ID=4;**



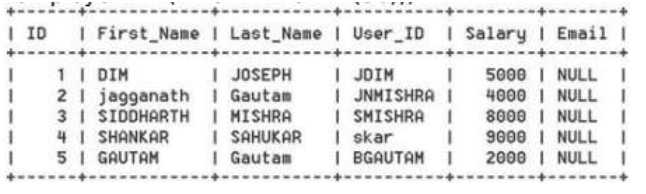
*2. Modify the last name of employees to Gautam where salary<5000.*

**UPDATE employe SET Last\_Name=’Gautam’ WHERE Salary;**



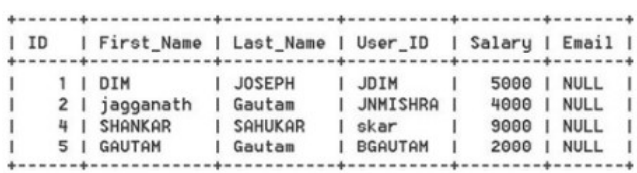
*3. Add column Email of data type VARCHAR to the table.*

**ALTER TABLE employe ADD(Email VARCHAR(30));**



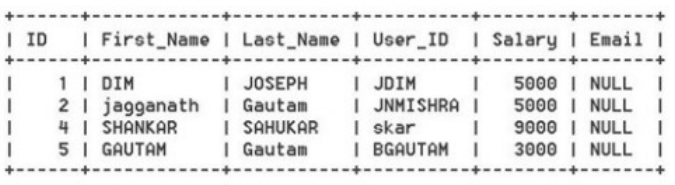
*4. Delete the employee record having first name as SIDDHARTH.*

**DELETE FROM employe WHERE First\_Name=’SIDDHARTH’;**



*5. Modify the salary and increases it by 1000, for all who get salary less than 5000.*

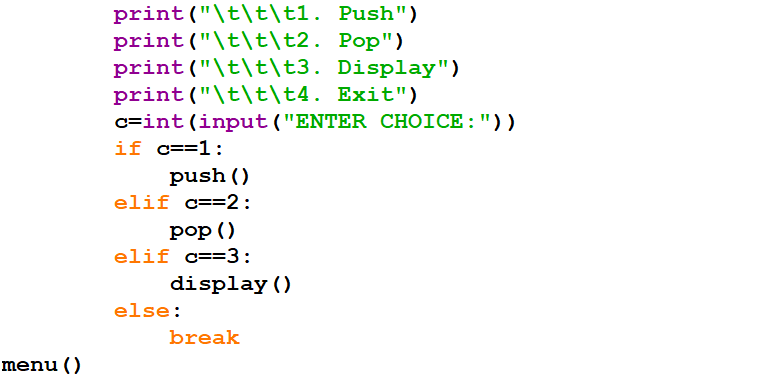
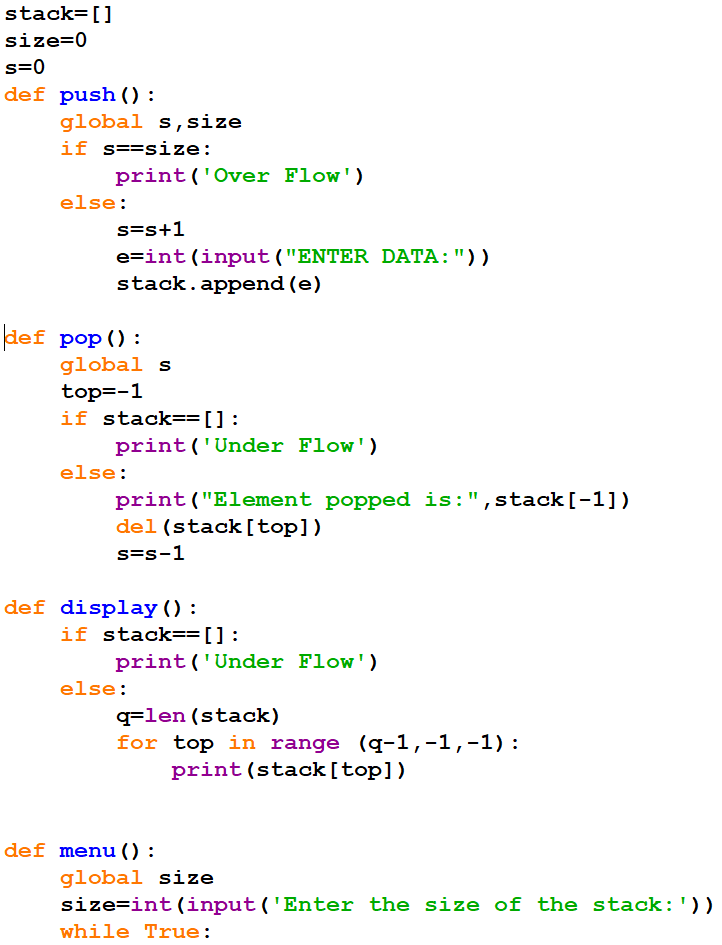
**UPDATE employe SET Salary =Salary+1000 WHERE Salary<5000;**



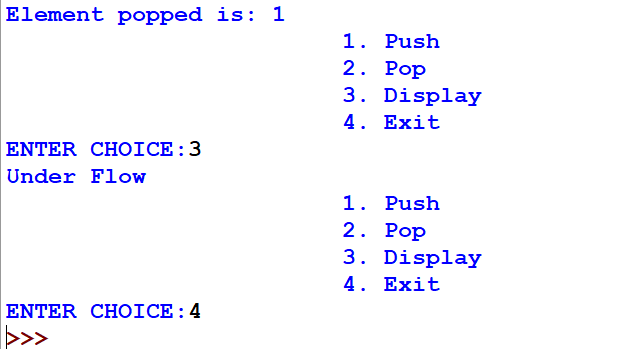
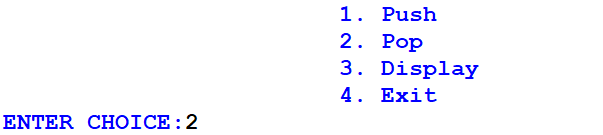
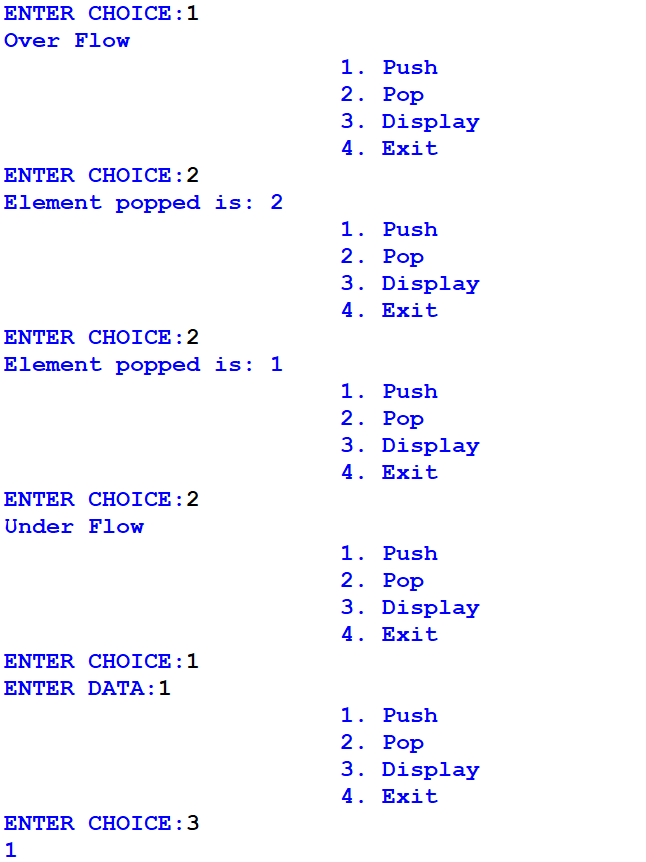
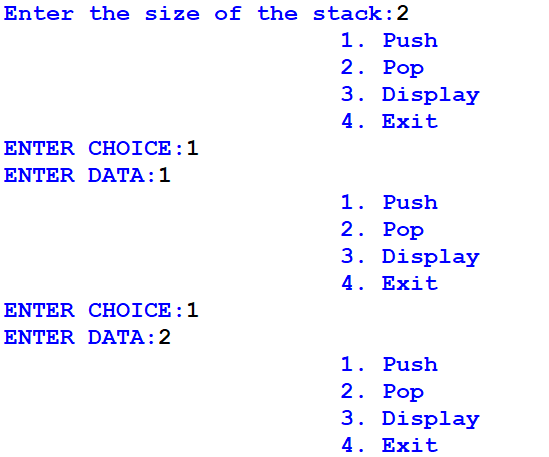
17. MYSQL CONNECTIVITY

18. STACKS

**INPUT**

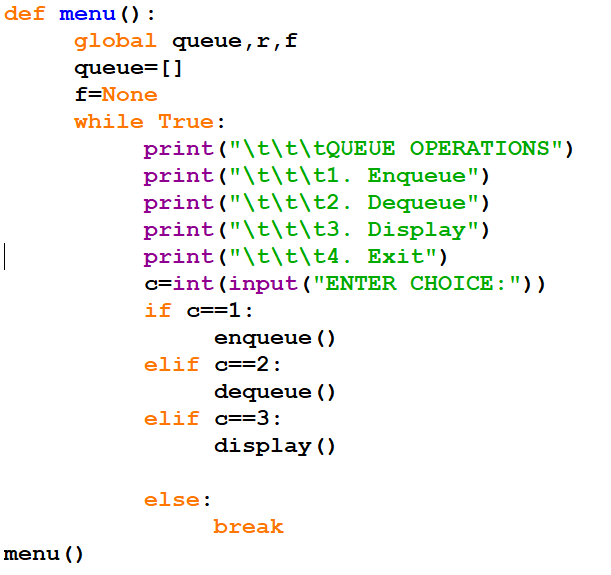
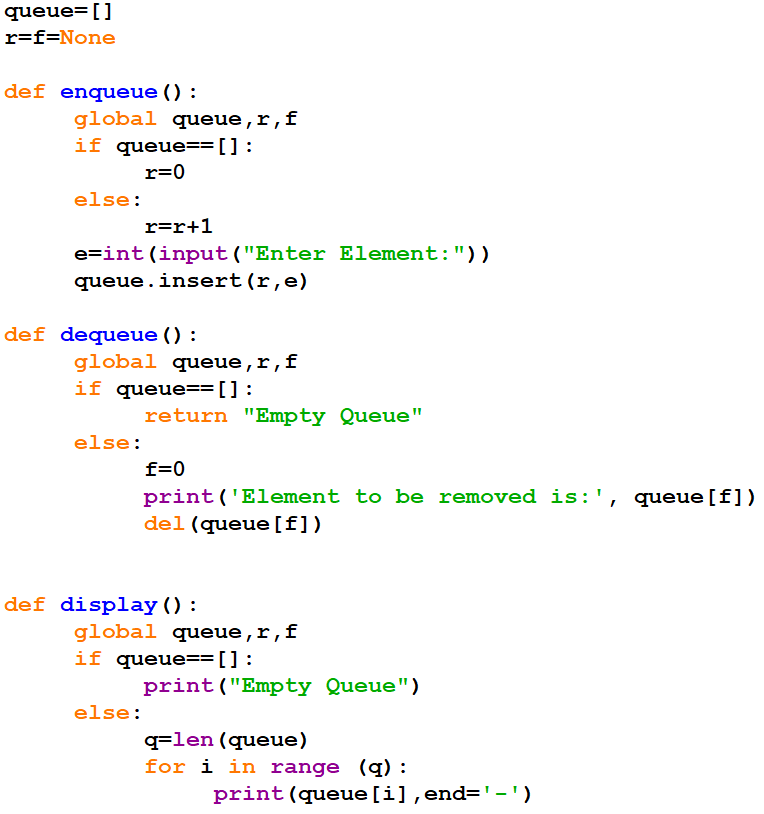


**OUTPUT**

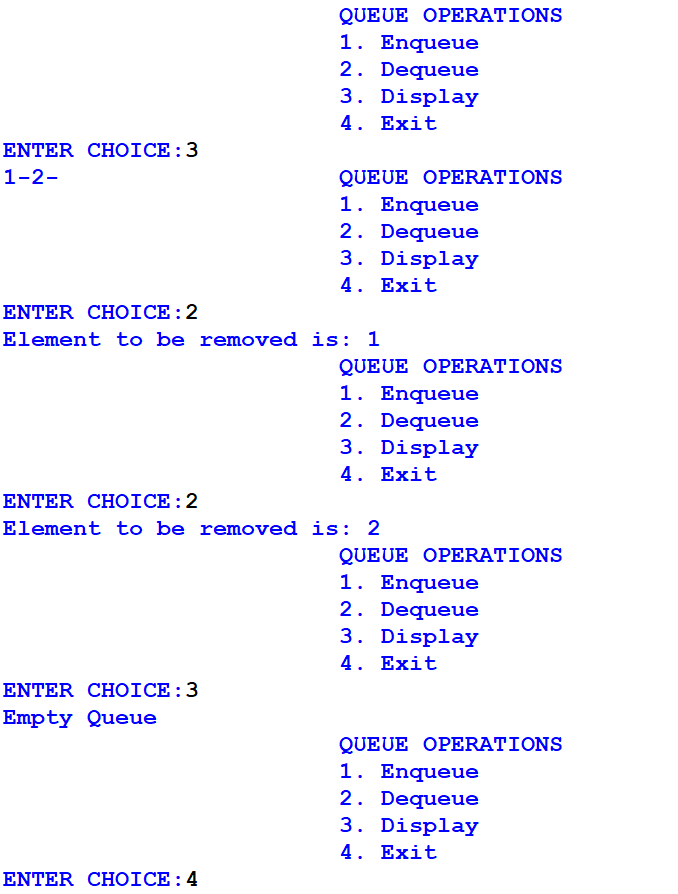
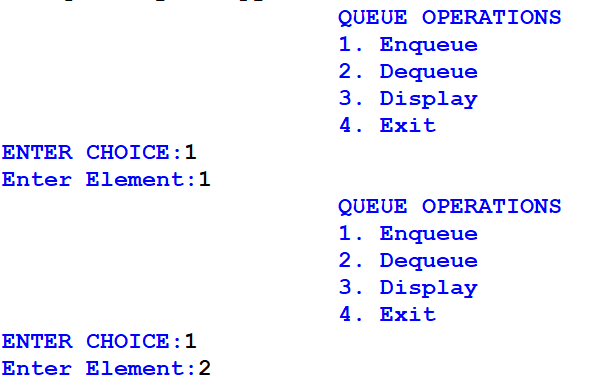


19. QUEUES

**INPUT**



**OUTPUT**



20. ART INTEGRATION PROJECT

**INPUT**

import pymysql

import matplotlib.pyplot as plt

from tkinter import \*

def create\_table():

    db=pymysql.connect(host="localhost",user="root",passwd=rootpwd,db=database)

    cur=db.cursor()

    cur.execute("create table stu(Roll int,Name char(20),Class char(5),English int, Physics int, Chemistry int, Maths int, Computers int, Percentage char(6),Grade char(2), Remark char(4));")

    db.commit()

    cur.close()

    db.close()

def add\_record\_screen():

    global mainframe

    mainframe.destroy()

    mainframe = Frame(root,width=1100,height=600,bg="#111")

    mainframe.grid\_propagate(0)

    mainframe.pack()

    def add\_record():

        roll=int(rolle.get())

        name=namee.get()

        clas=clase.get()

        eng=int(enge.get())

        mat=int(mate.get())

        cs=int(cse.get())

        chem=int(cheme.get())

        phy=int(phye.get())

        total = eng+mat+phy+chem+cs

        perc = round(total/500 \* 100,2)

        if perc > 33 : rem = 'PASS'

        else: rem = 'FAIL'

        if perc > 90: grade = 'A1'

        elif perc > 80: grade = 'A2'

        elif perc > 70: grade = 'B1'

        elif perc > 60: grade = 'B2'

        elif perc > 50: grade = 'C1'

        elif perc > 40: grade = 'C2'

        elif perc > 33: grade = 'D'

        else: grade = 'F'

        perc = str(perc)+"%"

        db=pymysql.connect(host="localhost",user="root",passwd=rootpwd,db=database)

        cur=db.cursor()

        cur.execute(f"insert into stu values({roll},\"{name}\",\"{clas}\",{eng},{phy},{chem},{mat},{cs},\"{perc}\",\"{grade}\",\"{rem}\");")

        db.commit()

        cur.close()

        db.close()

        rolle.delete(0,'end')

        namee.delete(0,'end')

        clase.delete(0,'end')

        enge.delete(0,'end')

        phye.delete(0,'end')

        cheme.delete(0,'end')

        mate.delete(0,'end')

        cse.delete(0,'end')

    Label(mainframe,bg="#111",fg="#fff",text='Roll No.').grid(row=1,column=1)

    Label(mainframe,bg="#111",fg="#fff",text='Name').grid(row=2,column=1)

    Label(mainframe,bg="#111",fg="#fff",text='Class').grid(row=3,column=1)

    Label(mainframe,bg="#111",fg="#fff",text='English').grid(row=4,column=1)

    Label(mainframe,bg="#111",fg="#fff",text='Physics').grid(row=5,column=1)

    Label(mainframe,bg="#111",fg="#fff",text='Chemistry').grid(row=6,column=1)

    Label(mainframe,bg="#111",fg="#fff",text='Mathematics').grid(row=7,column=1)

    Label(mainframe,bg="#111",fg="#fff",text='Comuper Science').grid(row=8,column=1)

    rolle = Entry(mainframe)

    rolle.grid(row=1,column=2)

    namee = Entry(mainframe)

    namee.grid(row=2,column=2)

    clase = Entry(mainframe)

    clase.grid(row=3,column=2)

    enge = Entry(mainframe)

    enge.grid(row=4,column=2)

    phye = Entry(mainframe)

    phye.grid(row=5,column=2)

    cheme = Entry(mainframe)

    cheme.grid(row=6,column=2)

    mate = Entry(mainframe)

    mate.grid(row=7,column=2)

    cse = Entry(mainframe)

    cse.grid(row=8,column=2)

    Button(mainframe,text="Back",command=Menu).grid(row=9,column=1)

    Button(mainframe,text="Submit",command=add\_record).grid(row=9,column=2)

def display():

    global box

    db=pymysql.connect(host="localhost",user="root",passwd=rootpwd,db=database)

    cur=db.cursor()

    rows=cur.execute("select \* from stu;")

    rec=cur.fetchall()

    records = """

+----------+-------------------------+----------+----------+----------+----------+----------+----------+----------+----------+----------+

|Roll No.  |Name                     |Class     |English   |Physics   |Chemistry |Maths     |CS        |Percentage|Grade     |Remarks   |

+----------+-------------------------+----------+----------+----------+----------+----------+----------+----------+----------+----------+

"""

    for i in rec:

        for j in range(len(i)):

            if j==0: records+="|"

            if j==1: records += "{0:<25}|".format(i[j])

            else:  records += "{0:<10}|".format(i[j])

        records+="\n+----------+-------------------------+----------+----------+----------+----------+----------+----------+----------+----------+----------+\n"

    cur.close()

    db.close()

    box.configure(state='normal')

    box.insert('end', records)

    box.configure(state='disabled')

def display\_graph():

    db=pymysql.connect(host="localhost",user="root",passwd=rootpwd,db=database)

    cur=db.cursor()

    rows=cur.execute("select \* from stu;")

    all\_records=cur.fetchall()

    avg\_eng=avg\_phy=avg\_chem=avg\_maths=avg\_cs=0

    for i in all\_records:

        avg\_eng+=i[3]

        avg\_phy+=i[4]

        avg\_chem+=i[5]

        avg\_maths+=i[6]

        avg\_cs+=i[7]

    avg\_eng/=rows

    avg\_phy/=rows

    avg\_chem/=rows

    avg\_maths/=rows

    avg\_cs/=rows

    bg2=[avg\_eng,avg\_phy,avg\_chem,avg\_maths,avg\_cs]

    try: rr=int(roll\_no.get())

    except:

        print('Please enter roll no.')

        return

    aa=f"select \* from stu where roll='{rr}';"

    rows=cur.execute(aa)

    rec=cur.fetchall()

    bg1=[]

    x=["English","Physics","Chemistry","Mathematics","Computer Science"]

    barWidth = 0.1

    for i in rec:

        bg1.append(i[3])

        bg1.append(i[4])

        bg1.append(i[5])

        bg1.append(i[6])

        bg1.append(i[7])

    r1 = [0,1,2,3,4]

    r2 = [i + barWidth for i in r1]

    plt.bar(r1,bg2,width=0.1,label="Class Average")

    plt.bar(r2,bg1,width=0.1,label="Student")

    plt.xlabel('group', fontweight='bold')

    plt.xticks([r + barWidth for r in range(len(bg1))], ['English', 'Physics', 'Chemistry', 'Maths', 'CS'])

    plt.legend()

    plt.show()

    cur.close()

    db.close()

def Menu():

    global mainframe,box,roll\_no

    mainframe.destroy()

    mainframe = Frame(root,width=1100,height=600,bg="#111")

    mainframe.grid\_propagate(0)

    mainframe.pack()

    Label(mainframe,text="Menu",bg="#111",fg="#fff",font=('serif',25)).grid(row=1,column=1)

    Button(mainframe,text="Create Table",command=create\_table).grid(row=2,column=1)

    Button(mainframe,text="Add Record",command=add\_record\_screen).grid(row=3,column=1)

    Button(mainframe,text="Display All Records",command=display).grid(row=4,column=1)

    Button(mainframe,text="Display Bar Graph Student Wise",command=display\_graph).grid(row=5,column=1)

    Button(mainframe,text="Exit",command=quit).grid(row=6,column=1)

    roll\_no = Entry(mainframe)

    roll\_no.grid(row=7,column=1)

    box=Text(mainframe,width=137,height=25,bg='#333',fg='#fff',state='disabled')

    box.grid(row=8,column=1)

def connect(a1,a2):

    global database,rootpwd

    rootpwd = a1

    database = a2

    Menu()

def connect\_screen():

    e1 = Entry(mainframe,show="\*")

    e2 = Entry(mainframe)

    Label(mainframe,text="Enter root@localhost Password",bg="#111",fg="#fff").grid(row=1,column=1)

    e1.grid(row=1,column=2)

    Label(mainframe,text="Enter name of database to be used",bg="#111",fg="#fff").grid(row=2,column=1)

    e2.grid(row=2,column=2)

    Button(mainframe,text="Submit",command=lambda: connect(e1.get(),e2.get()) ).grid(row=3,column=1,columnspan=2)

root = Tk()

root.geometry('1100x600')

mainframe = Frame(root,width=1100,height=600,bg="#111")

mainframe.grid\_propagate(0)

mainframe.pack()

connect\_screen()

root.mainloop()

**OUTPUT:**

